What is claimed is:

1. A digital computer comprising:

at least one processor and a random access memory, said at least one processor being coupled to the random access memory for access to the random access memory;

said at least one processor being programmed with a watchdog thread that is normally executed by said at least one processor during a watchdog grace period, and the digital computer being programmed to shut down and restart when said at least one processor fails to execute the watchdog thread during the watchdog grace period; and

said at least one processor being programmed with a periodic interrupt routine that is executed by said at least one processor for checking whether said at least one processor holds any spinlocks, and upon finding that said at least one processor does not hold any spinlocks, for executing the watchdog thread, and upon finding that said at least one processor holds at least one spinlock, for deferring execution of the watchdog thread until said at least one processor releases all of the spinlocks held by said at least one processor.

2. The digital computer as claimed in claim 1, wherein said at least one processor is programmed for deferring execution of the watchdog thread by setting an indicator upon finding that said at least one processor holds at least one spinlock, and by checking the indicator when said at least one processor releases all of the spinlocks held by said at least one processor, and upon checking the indicator and finding that the indicator is set, preempting execution of a current thread for execution of the watchdog thread.

3. The digital computer as claimed in claim 1, which includes multiple processors coupled to the random access memory for shared access to the random access memory, the multiple processors being programmed for obtaining spinlocks for access to the memory, and the digital computer being programmed for maintaining a count for each of the processors of the number of spinlocks held by said each of the processors.

4. The digital computer as claimed in claim 3, wherein said each of the processors is programmed for decrementing the count for said each of the processors upon releasing a spinlock and checking whether the count for said each of the processors has been decremented to zero, and upon finding that the count for said each of the processors has been decremented to zero, for preempting execution of a current thread for execution of the watchdog thread.

5. The digital computer as claimed in claim 4, wherein said each of the processors is programmed for preempting execution of the current thread for execution of the watchdog thread by removing a thread from a respective run queue for said each of the processors, placing the current thread on the respective run queue for said each of the processors, and transferring execution to the thread removed from the respective run queue for said each of the processors.

6. The digital computer as claimed in claim 1, wherein said at least one processor is programmed for keeping a count of the number of times that execution of the watchdog thread has been deferred for said at least one of the processors during the watchdog grace period.

7. A digital computer comprising:

multiple processors; and

a random access memory coupled to the processors for shared access to the random access memory;

each of the processors being programmed for normally executing a watchdog thread during a watchdog grace period;

the digital computer being programmed with a system watchdog routine for detecting improper program execution when any of the processors fails to execute the watchdog thread during the watchdog grace period; and

said each of the processors being programmed with a periodic interrupt routine for checking whether said each of the processors holds any spinlocks, and upon finding that said each of the processors does not hold any spinlocks, for preempting execution of an interrupted current thread to permit execution of the watchdog thread, and upon finding that said each of the processors holds at least one spinlock, for deferring preemption of execution of the current thread for enabling execution of the watchdog thread once said each of the processors releases all of the spinlocks held by said each of the processors.

8. The digital computer as claimed in claim 7, wherein said each of the processors is programmed for deferring preemption of execution of the current thread upon finding that said each of the processors holds at least one spinlock by setting a respective indicator for said each of the processors upon finding that said each of the processors holds at least one spinlock, and for checking the respective indicator for said each of the processors when said each of the processors releases all of the spinlocks held by said at least one of the processors, and upon checking the respective indicator for said each of the processors and finding that the respective indicator for said each of the processors is set, for preempting execution of the current thread for enabling execution of the watchdog thread.

9. The digital computer as claimed in claim 7, wherein the digital computer is programmed to pass a single instance of the watchdog thread among the processors.

10. The digital computer as claimed in claim 7, wherein the watchdog thread is programmed to set a processor status bit for said each of the processors when the watchdog thread is executed by said each of the processors.

11. The digital computer as claimed in claim 7, wherein the digital computer is programmed to execute the periodic interrupt routine in synchronism with a thread handler interrupt.

12. The digital computer as claimed in claim 7, wherein the digital computer is programmed to maintain a count for said each of the processors of spinlocks upon said each of the processors.

13. The digital computer as claimed in claim 12, wherein said each of the processors is programmed for decrementing the count for said each of the processors when releasing a spinlock and for checking whether said each of the processors has decremented the count for said each of the processors to zero, and upon finding that said each of the processors has decremented the count for said each of the processors to zero, for preempting execution of the current thread of said each of the processors for execution of the watchdog thread.

14. The digital computer as claimed in claim 12, wherein the periodic interrupt routine is programmed to terminate for said each of the processors when the count for said each of the processors of spinlocks upon said each of the processors is zero and the current thread of said at least one of the processors is a real-time thread or an idle thread.

15. The digital computer as claimed in claim 12, wherein the periodic interrupt routine is programmed to preempt execution of a current thread of said each of the processors in favor of execution of the watchdog thread by said each of the processors when the count for said each of the processors of spinlocks upon said each of the processors is zero and the current thread of said each of the processors is neither a real-time thread nor an idle thread.

16. The digital computer as claimed in claim 7, wherein said each of the processors is programmed for keeping a respective count of the number of times that preemption is deferred for said each of the processors during the watchdog grace period.

17. A method of operating a digital computer having at least one processor coupled to random access memory for accessing the random access memory, said method comprising said at least one processor normally executing a watchdog thread during a watchdog grace period, and said at least one processor being shutdown and restarted upon failing to execute the watchdog thread during the watchdog grace period; and

said at least one processor being periodically interrupted for checking whether said at least one processor holds any spinlocks, and upon finding that said at least one processor does not hold any spinlocks, executing the watchdog thread, and upon finding that said at least one processor holds at least one spinlock, deferring execution of the watchdog thread until said at least one processor releases all of the spinlocks held by said at least one processor.

18. The method as claimed in claim 17, which includes said at least one processor deferring execution of the watchdog thread by setting an indicator upon finding that said at least one processor holds at least one spinlock, and by checking the indicator when said at least one processor releases all of the spinlocks held by said at least one processor, and upon checking the indicator and finding that the indicator is set, preempting execution of a current thread for execution of the watchdog thread.

19. The method as claimed in claim 17, wherein the digital computer includes multiple processors coupled to the random access memory for shared access to the random access memory, and wherein the method includes the digital computer maintaining a count for each of the processors of the number of spinlocks held by said each of the processors.

20. The method as claimed in claim 19, wherein said each of the processors decrements the count for said each of the processors upon releasing a spinlock and checks whether the count for said each of the processors has been decremented to zero, and upon finding that the count for said each of the processors has been decremented to zero, said each of the processors preempts execution of a current thread for execution of the watchdog thread.

21. The method as claimed in claim 20, wherein said each of the processors preempts execution of the current thread for execution of the watchdog thread by removing a thread from a respective run queue for said each of the processors, placing the current thread on the respective run queue for said each of the processors, and transferring execution to the thread removed from the respective run queue.

22. The method as claimed in claim 17, wherein said at least one of the processors keeps a count of the number of times that execution of the watchdog thread is deferred during the watchdog grace period for said at least one of the processors.

23. The method as claimed in claim 22, which includes performing a diagnostic procedure after a shutdown and restart of the digital computer, the diagnostic procedure inspecting saved processor status information including the respective count of the number of times that execution of the watchdog thread is deferred during the watchdog grace period for said at least one of the processors in order to determine whether the shutdown and restart of the digital computer is likely to have been caused by said at least one of the processors failing to have released any spinlocks during the watchdog grace period.

24. A method of operating a digital computer including multiple processors and a random access memory coupled to the processors for shared access to the random access memory, said method comprising each of the processors normally executing a watchdog thread during a watchdog grace period, and the digital computer being shutdown and restarted upon any of the processors failing to execute the watchdog thread during the watchdog grace period; and

said each of the processors being periodically interrupted, and upon being periodically interrupted, checking whether said each of the processors holds any spinlocks, and upon finding that said each of the processors does not hold any spinlocks, preempting execution of an interrupted current thread to permit execution of the watchdog thread, and upon finding that said each of the processors holds at least one spinlock, deferring preemption of execution of the current thread for enabling execution of the watchdog thread once said each of the processors releases all of the spinlocks held by said each of the processors.

25. The method as claimed in claim 24, which includes said each of the processors deferring preemption of execution of the current thread upon finding that said each of the processors holds at least one spinlock by setting a respective indicator for said each of the processors upon finding that said each of the processors holds at least one spinlock, and by checking the respective indicator for said each of the processors when said each of the processors releases all of the spinlocks held by said at least one of the processors, and upon checking the respective indicator for said each of the processors and finding that the respective indicator for said each of the processors is set, preempting execution of the current thread for enabling execution of the watchdog thread.

26. The method as claimed in claim 24, which includes passing execution of a single instance of the watchdog thread among the processors.

27. The method as claimed in claim 24, which includes the watchdog thread setting a processor status bit for said each of the processors when the watchdog thread is executed by said each of the processors.

28. The method as claimed in claim 24, which includes the digital computer executing a thread handler routine in synchronism with the periodic interrupt.

29. The method as claimed in claim 24, which includes the digital computer maintaining a count for said each of the processors of spinlocks upon said each of the processors.

30. The method as claimed in claim 29, which includes said each of the processors decrementing the count for said each of the processors when releasing a spinlock and checking whether said each of the processors has decremented the count for said each of the processors to zero, and upon finding that said each of the processors has decremented the count for said each of the processors to zero, preempting execution of the current thread of said each of the processors for execution of the watchdog thread.

31. The method as claimed in claim 29, which includes terminating execution by said each of the processors of an interrupt routine for handling the periodic interrupt when the count for said each of the processors of spinlocks upon said each of the processors is zero and the current thread of said each of the processors is a real-time thread or an idle thread.

32. The method as claimed in claim 29, wherein said each of the processors responds to the periodic interrupt by preempting execution of the current thread of said each of the processors in favor of execution of the watchdog thread by said each of the processors when the count for said each of the processors of spinlocks upon said each of the processors is zero and the current thread of said each of the processors is neither a real-time thread nor an idle thread.

33. The method as claimed in claim 24, wherein said each of the processors keeps a respective count of the number of times that preemption is deferred for said each of the processors during the watchdog grace period.

34. The method as claimed in claim 33, which includes performing a diagnostic procedure after a shutdown and restart of the digital computer, the diagnostic procedure inspecting saved processor status information including the respective count of the number of times that preemption is deferred during the watchdog grace period for at least one of the processors in order to determine whether the shutdown and restart of the digital computer is likely to have been caused by said at least one of the processors failing to have released any spinlocks during the watchdog grace period.